

REMARKS

Claims 1, 3, 6-10, 12-17, 20-22, 31-33, 35-39, and 42-44 were rejected by the Examiner. In response, Applicants now amend claims 16 and 38. Accordingly, claims 1, 3, 6-10, 12-17, 20-22, 31-33, 35-39, and 42-44 remain pending in the Application.

RESPONSE TO ARGUMENTS/AMENDMENTS

In “Response to Arguments/Amendments” item 2 on page 2 of the above-identified final Office Action, the Examiner acknowledged Applicants’ offer to submit a terminal disclaimer upon issuance of either U.S. Patent Application No. 10/784,492 or the instant application. As of the mailing date of this communication, neither application has issued.

In “Response to Arguments/Amendments” item 3 on page 2 of the above-identified final Office Action, the Examiner withdrew the rejection of claim 1 under §112, second paragraph. Applicants thank the Examiner for withdrawing the rejection.

In “Response to Arguments/Amendments” item 4 on page 2 of the above-identified final Office Action, the Examiner responded to arguments offered by Applicants on pages 14-16 of Applicants’ December 4, 2007 response and stated that claims 16, 31, and 38 do not recite the compiler recited by claim 1.

First, Applicants note that claim 31 does in fact recite a compiler (see lines 10 and 14 of claim 31). With regard to claims 16 and 38, Applicants have amended claims 16 and 38 to recite the compiler recited by claims 1 and 31.

Second, Applicants respectfully disagree with the remarks made by the Examiner in the response to Applicants’ prior arguments made on pages 2-4 of the final Office Action.

More specifically, on pages 2-3, the Examiner states that WebLogic discloses declarative annotations, that Ringseth teaches the inclusion of declarative annotations within source code, and that Applicants do not disagree with the Examiner regarding those teachings. The Examiner then states that Monson-Haeft teaches “declarative annotations which include start, continue, and finish methods.” Applicants respectfully disagree with this last contention.

As Applicants acknowledge, Monson-Haeftl may arguably be read as suggesting start, continue, and finish methods as the “identified method” of the “web service logic” recited by claim 1. The passages of Monson-Haeftl cited by the Examiner 7.4.2 and 10.6.3.2 do teach at least a start method of a stateful bean. But they do not disclose declarative annotations or, consequently, that “declarative annotations indicate to the compiler whether the identified method is a start method...”, as claimed by claim 1. While WebLogic and Ringseth do teach declarative annotations included in source code, the mere combination of that teaching with source code that includes a start method (as taught by Monson-Haeftl) simply does not give rise to a teaching or suggestion of “declarative annotations indicate to the compiler whether the identified method is a start method....” For the combination of references to suggest this recitation of claim 1, there must at least be some reason or benefit that would cause one of ordinary skill to modify the combination of references to have the declarative annotations indicate to a compiler whether an identified method is a start method, etc. But as Applicants noted on pages 15-16 of the December 4, 2007 response, there is no such reason; in fact, the combination of references teaches away from such a modification.

As stated in the December 4, 2007 response: “WebLogic, Ringseth, and Monson-Haeftl arguably teach away from the need to identify the create(), etc. methods to a compiler since the mere act of compiling the methods and bean logic will result in the desired bean to maintain state. In amended claim 1, in contrast, the identified method is indicated to be a start, etc. method precisely because compiling the method and the logic alone (i.e., without declarative annotations) will not result in the persistent component of claim 1 to maintain state. Thus, by teaching a method to actually instantiate a persistent component (i.e., bean), WebLogic, Ringseth, and Monson-Haeftl teach away from the need for the declarative annotations to indicate to a compiler that a method is a start method, the annotations causing the compiler to create a persistent component.”

Accordingly, WebLogic, Ringseth, and Monson-Haeftl do not suggest “declarative annotations indicate to the compiler whether the identified method is a start method...” because there would be absolutely no need or reason for indicating such a thing to the compiler. In fact, such a modification would cause wasteful processing of unnecessary annotations. Thus, the combined references certainly teach away from the “declarative

annotations indicate to the compiler whether the identified method is a start method...” recited by claim 1.

On page 3 of the final Office Action, the Examiner then responds to “applicant’s argument that there is no suggestion to combine the references.” Applicants, however, made no such argument. The “suggestion” argument referred to by the Examiner at the bottom of page 15 of Applicants’ December 4, 2007 response was an argument that the references failed to teach or suggest each and every limitation of Applicants claims, not an argument that there is no suggestion to combine the references. Applicants did not reach or address the “suggestion/motivation to combine” issue because Applicants concluded that the combined references simply do not teach or suggest the recitations of claim 1. Thus, Applicants respectfully maintain, regardless of whether there is motivation to combine the references, that the references do not suggest each and every element of claim 1. Applicants do not admit that any such motivation to combine exists, but merely note that such existence need not be addressed at this time.

Further on pages 3-4 of the final Office Action, the Examiner states that Applicants fail to show that the cited references teach away from the invention because Applicants did not show where the references “actually discourage using start, continue, or finish methods, or otherwise discourage the use of stateful components.” Applicants in response note that the Examiner has not actually addressed Applicants’ argument. Applicants did not argue that the cited references teach away from start, etc. methods or from stateful components. Rather, Applicants argued that the references teach away from having “declarative annotations indicate to the compiler whether the identified method is a start method...” While no portion of the references explicitly discourages the use of declarative annotations for indicating to a compiler, the references do implicitly teach away from such a use of declarative annotations, as Applicants discuss above and below.

Also, on page 4 of the final Office Action, the Examiner addresses Applicants’ previous arguments with regard to claim 15. In those arguments, Applicants noted that the cited references fail to teach or suggest “wherein the one or more declarative annotations further cause the compiler to generate a unique identifier to identify a specific conversational instance of the external service.” In response, the Examiner notes that Monson-Haefel teaches “instantiation of a conversation and return of an identifier upon invocation of a start

method.” The Examiner then equates this teaching of Monson-Haefel to the recitations of claim 15. Applicants respectfully disagree. The mere exchange of a unique identifier to identify a conversation in no way suggests generation of the unique identifier by a compiler in response to the presense of declarative annotations. Applicants respectfully request that the Examiner withdraw the rejection of claim 15 or point to a portion of one of the references that suggests generation of the unique identifier by a compiler in response to the presense of declarative annotations, as claimed by claim 15.

DOUBLE-PATENTING

In “Double Patenting” item 6 on page 5 of the above-identified final Office Action, claims 1, 3, 6-10, 12-17, 20-22, 31-33, 35-39, and 42-44 have been provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claim 1-8, 19-23, 26, 27, 31-36, 38, 39, 43, and 44 of copending Application No. 10/784,492 (hereinafter ‘492) which was filed after the instant application.

As mentioned above, Applicants will, upon issuance of either ‘492 or the instant application, submit the necessary Terminal Disclaimer for the remaining application. Thus, there will be no double patenting.

CLAIM REJECTIONS UNDER 35 U.S.C.-, § 103

1. In “Claim Rejections – 35 USC § 103,” item 12 on page 6 of the above-identified Office Action, claims 1, 6-8, 10, 16, 22, 31, 38, and 44 have been rejected as being unpatentable over “Using WebLogic Enterprise JavaBeans” by BEA Systems (hereinafter “WebLogic”) in view of U.S. Patent Publication Number 2003/0014733 to *Ringseth et al.* (hereinafter “Ringseth”) in view of “Enterprise JavaBeans” by Monson-Haefel (hereinafter “Monson-Haefel”) under 35 U.S.C. § 103(a).

Amended claim 1 teaches a “method of specifying a stateful web service comprising:
first facilitating, by an integrated development environment of a computing device, a user in providing a source code representation of at least a portion of web service logic, the logic including one or more methods;

second facilitating, by the integrated development environment of the computing device, the user in identifying one of said one or more methods to be exposed as part of the stateful web service; and

in response to user input, automatically specifying, by the integrated development environment of the computing device, one or more declarative annotations within the source code representation, the declarative annotations, when recognized by a compiler through analysis of the represented portion of web service logic which includes the declarative annotations, causing the compiler to generate one or more persistent components to maintain conversational state related to the identified method;

wherein the one or more declarative annotations indicate to the compiler whether the identified method is at least one of a start method, a continue method, or a finish method, wherein the start method applies to start of a stateful conversation between a client and the web service, the continue method applies to continuation of an ongoing stateful conversation between a client and the web service, and the finish method applies to completion of an ongoing stateful conversation between a client and the web service.”

In rejecting claim 1, the Examiner again pointed to numerous sections of WebLogic, those sections discussed in depth with respect to the elements of claim 1 in prior responses. WebLogic simply teaches a method for specifying, by a developer, a conventional EJBBean and deployment descriptor which may serve as a stateful component of a web service. In prior rejections, the Examiner equated the deployment descriptor to the declarative annotations of claim 1. In response, Applicant previously amended claim 1 to indicate that the annotations were specified within the WebLogic, thus distinguishing over WebLogic, in which the deployment descriptors are found in an external file. In response to that amendment, the Examiner cited Ringseth as curing the deficiencies of WebLogic. Ringseth teaches that a developer may specify declarative annotations, in the form of Visual C++ attributes, within web service source code, the attributes to be recognized by a compiler as specifying objects for handling SOAP messages.

In response to the Examiner's rejection, Applicants amended claim 1 to recite: "wherein the one or more declarative annotations indicate to the compiler whether the identified method is at least one of a start method, a continue method, or a finish method..." in their December 4, 2007 response.

In the above-identified final Office Action, the Examiner rejected claim 1 as being unpatentable over WebLogic, Ringseth, and Monson-Haefel. Monson-Haefel is a text on Enterprise JavaBeans. In Part 7.4 of that text, Monson-Haefel discusses the life-cycle of a stateful session bean. Monson-Haefel notes that a stateful bean may have three states: does not exist, method-ready, and passivated. To instantiate a bean and enter a method-ready state, a client may invoke a create() method of a bean. The Examiner equates the create() method to a start method. The Examiner also cites ejbActive() and ejbRemove() as continue and finish methods, respectively. Because the start, etc. method of claim 1 is the "identified method" (i.e., the method belonging to the source code representation of the at least a portion of web service logic recited by claim 1), the Examiner is implicitly equating the bean logic to which the create(), etc. methods belong to the source code representation. This is consistent with current and previous rejections of claim 1, in which the Examiner reads the bean logic of WebLogic on the source code representation.

But even assuming that bean logic is capable of reading on the source code representation and that create(), etc. is capable of reading on the identified start, etc. method (points which Applicants do not concede), WebLogic, Ringseth, and Monson-Haefel simply do not teach or suggest that the create(), etc. methods are indicated to a compiler as start, etc. methods by declarative annotations, as recited by amended claim 1. While Ringseth does describe declarative annotations, those annotations simply instruct a compiler as to the SOAP message handling capabilities desired for web service logic. They do not "indicate to the compiler whether the identified method is at least one of a start method ... wherein the start method applies to start of a stateful conversation between a client and the web service." Further, there is no suggestion in WebLogic, Ringseth, and Monson-Haefel to extend the declarative annotations of Ringseth to the create(), etc. methods of the bean logic of Monson-Haefel. In fact, WebLogic, Ringseth, and Monson-Haefel arguably teach away from the

need to identify the create(), etc. methods to a compiler since the mere act of compiling the methods and bean logic will result in the desired bean to maintain state. In claim 1, in contrast, the identified method is indicated to be a start, etc. method precisely because compiling the method and the logic alone (i.e., without declarative annotations) will not result in the persistent component of claim 1 to maintain state. Thus, by teaching a method to actually instantiate a persistent component (i.e., bean), WebLogic, Ringseth, and Monson-Haefel teach away from the need for the declarative annotations to indicate to a compiler that a method is a start method, the annotations causing the compiler to create a persistent component.

Accordingly, claim 1 is patentable over WebLogic, Ringseth, and Monson-Haefel under §103.

Claims 16, 31, and 38 recite limitations similar to those of claim 1. Accordingly, claims 16, 31, and 38 are patentable over WebLogic, Ringseth, and Monson-Haefel for at least the same reasons.

Claims 6-8, 10, 22, and 44 depend from claims 1, 16, and 38, incorporating their limitations, respectively. Accordingly, for at least the same reasons, claims 6-8, 10, 22, and 44 are patentable over WebLogic, Ringseth, and Monson-Haefel under §103.

2. In “Claim Rejections – 35 USC § 103,” item 9 on page 16 of the above-identified final Office Action, claim 3 has been rejected as being unpatentable over WebLogic, Ringseth, and Monson-Haefel, as applied to claim 1 above, and further in view of “EJBDoclet,” December 21, 2000, by dreamBean Software (hereinafter “EJBDoclet”).

EJBDoclet is proffered for the teaching of “the one or more declarative annotations are specified within a comment field preceding the identified method”. EJBDoclet does not cure the deficiencies of WebLogic, Ringseth, and Monson-Haefel. Accordingly, claim 1 remains patentable over WebLogic, Ringseth, Monson-Haefel, and EJBDoclet, alone or in combination, for at least the reasons given above. Claim 3 depends from claims 1,

incorporating its limitations. Accordingly, claim 3 is patentable over WebLogic, Ringseth, Monson-Haeftul, and EJBDoclet, alone or in combination, under §103(a).

3. In “Claim Rejections – 35 USC § 103,” item 10 on page 17 of the above-identified final Office Action, claims 9, 17, and 39 have been rejected as being unpatentable over WebLogic, Ringseth, and Monson-Haeftul as applied to claims 1 above, and further in view of prior art of record U.S. Patent 5,812,768 to Pagé, et al. (hereinafter “Pagé”).

Pagé was proffered for the teachings of “wherein the one or more declarative annotations indicate to the compiler whether the identified method is buffered, wherein if the identified method is buffered the compiler instantiates one or more queues to temporarily store one or more requests for the identified method”. Pagé does not cure the deficiencies of WebLogic, Ringseth, and Monson-Haeftul. Accordingly, claims 1, 16, and 38 remain patentable over WebLogic, Ringseth, Monson-Haeftul, and Pagé, alone or in combination, for at least the reasons given above. Claims 9, 17, and 39 depend from claims 1, 16, and 38, respectively, incorporating their limitations. Accordingly, claims 9, 17, and 39 are patentable over WebLogic, Ringseth, Monson-Haeftul, and Pagé, alone or in combination, under §103(a).

4. In “Claim Rejections – 35 USC § 103,” item 11 on page 18 of the above-identified final Office Action, claims 12 and 32 have been rejected as being unpatentable over WebLogic, Ringseth, and Monson-Haeftul, as applied to claims 1 and 31 above, and further in view of U.S. Patent 6,230,160 to Chan, et al. (hereinafter “Chan”).

Chan was proffered for the teachings of “wherein said user input includes graphical manipulation of the identified method by the user via the integrated development environment”. Chan does not cure the deficiencies of WebLogic, Ringseth, and Monson-Haeftul. Accordingly, claims 1 and 31 remain patentable over WebLogic, Ringseth, Monson-Haeftul, and Chan, alone or in combination, for at least the reasons given above. Claims 12 and 32 depend from claims 1 and 31, incorporating their limitations. Accordingly,

claim 12 and 32 are patentable over WebLogic, Ringseth, Monson-Haefel, and Chan, alone or in combination, under §103(a).

5. In “Claim Rejections – 35 USC § 103,” item 12 on page 19 of the above-identified final Office Action, claims 13, 15, 20, 21, 35, 37, 42, and 43 have been rejected as being unpatentable over WebLogic, Ringseth, and Monson-Haefel as applied to claim 1 above, and further in view of the “Background of the Invention” section appearing on pages 1-3 of the originally filed specification (hereinafter “BOTI”).

BOTI was proffered for the teachings of “wherein the one or more declarative annotations cause the compiler to generate a proxy object designed to facilitate interaction by the web service with one of an external web service or client”. BOTI does not cure the deficiencies of WebLogic, Ringseth, and Monson-Haefel. Accordingly, claims 1, 16, 31, and 38 remain patentable over WebLogic, Ringseth, Monson-Haefel, and BOTI, alone or in combination, for at least the reasons given above. Claims 13, 15, 20, 21, 35, 37, 42, and 43 depend from claims 1, 16, 31, and 38, respectively, incorporating their limitations. Accordingly, claims 13, 20, 35, and 42 are patentable over WebLogic, Ringseth, Monson-Haefel, and BOTI, alone or in combination, under §103(a).

Additionally, WebLogic, Ringseth, Monson-Haefel, and BOTI do not teach or suggest the additional elements of claims 15, et seq., namely “wherein the one or more declarative annotations further cause the compiler to generate a unique identifier to identify a specific conversational instance of the external service”, as is claimed by claims 15 et seq. As support for the rejections of claims 15 et seq., the Examiner simply points to Monson-Haefel generally as teaching or suggesting the additional elements. Monson-Haefel does not, however, describe relations between a web service and an external service, much less assigning a unique identifier to a conversation with the external service, by a compiler, in response to declarative annotations. Rather, Monson-Haefel merely describes the behavior of stateful and stateless beans. No mention is made of a conversation identifier for a conversation with an external service. The only identifiers mentioned are identifiers of

conversations with clients. Thus, WebLogic, Ringseth, BOTI, and Monson-Haefel simply do not teach or suggest claims 15, 21, 37, and 43, for at least these additional reasons.

6. In “Claim Rejections – 35 USC § 103,” item 13 on page 21 of the above-identified final Office Action, claims 14 and 36 have been rejected as being unpatentable over WebLogic, Ringseth, Monson-Haefel, and BOTI as applied to claims 13 and 35 above, and further in view of Pagé.

Page was proffered for the teachings of “wherein the one or more declarative annotations further cause the compiler to route asynchronous responses from the external web service to code specified by a developer of the web service”. Pagé does not cure the deficiencies of WebLogic, Ringseth, Monson-Haefel, and BOTI. Accordingly, claims 13 and 35 remain patentable over WebLogic, Ringseth, Monson-Haefel, BOTI, and Pagé, alone or in combination, for at least the reasons given above. Claims 14 and 36 depend from claims 13 and 35, incorporating their limitations. Accordingly, claims 14 and 36 are patentable over WebLogic, Ringseth, Monson-Haefel, BOTI, and Pagé, alone or in combination, under §103(a).

7. In “Claim Rejections – 35 USC § 103,” item 14 on page 21 of the above-identified final Office Action, claim 33 has been rejected as being unpatentable over WebLogic, Ringseth, Monson-Haefel, and Chan as applied to claim 32 above, and further in view of dreamBean.

dreamBean does not cure the deficiencies of WebLogic, Ringseth, Monson-Haefel, and Chan. Accordingly, claim 32 remains patentable over WebLogic, Ringseth, Monson-Haefel, Chan, and dreamBean, alone or in combination, for at least the reasons given above. Claim 33 depends from claim 32, incorporating its limitations. Accordingly, claim 33 is patentable over WebLogic, Ringseth, Monson-Haefel, Chan, and dreamBean, alone or in combination, under §103(a).

CONCLUSION

In view of the foregoing, reconsideration and allowance of claims 1, 3, 6-10, 12-17, 20-22, 31-33, 35-39, and 42-44 are solicited. As a result of the amendments made herein, Applicant submits that claims 1, 3, 6-10, 12-17, 20-22, 31-33, 35-39, and 42-44 are in condition for allowance. Accordingly, a Notice of Allowance is respectfully requested. If the Examiner has any questions concerning the present paper, the Examiner is kindly requested to contact the undersigned at (206) 407-1513. If any fees are due in connection with filing this paper, the Commissioner is authorized to charge the Deposit Account of Schwabe, Williamson and Wyatt, P.C., No. 50-0393.

Respectfully submitted,
SCHWABE, WILLIAMSON & WYATT, P.C.

Date: April 23, 2008

by: /Robert C. Peck/
Robert C. Peck
Reg. No.: 56,826

Schwabe, Williamson & Wyatt, P.C.
Pacwest Center, Suites 1600-1900
1211 SW Fifth Avenue
Portland, Oregon 97222
Telephone: 503-222-9981